# User-Centered Design and Evaluation of Virtual Environments

Joseph L. Gabbard
*VPST*

Deborah Hix
*Virginia Polytechnic Institute and State University*

J. Edward Swan II
*Naval Research Laboratory*

**We present a methodology for ensuring the usability of virtual environments through user-centered design and evaluation.**

The ever-increasing power of computers and hardware rendering systems enables the creation of visually rich and perceptually realistic virtual environment (VE) applications. At the same time, comparatively little effort has gone into the user interaction components of VEs. Although usability engineering is a newly emerging facet of VE development, user-centered design and evaluation in VEs as a practice still lags far behind what's needed.

In this article we present a structured, iterative methodology for user-centered design and evaluation of VE user interaction. Figure 1 illustrates our basic technique. We recommend performing (1) user task analysis followed by (2) expert guidelines-based evaluation, (3) formative user-centered evaluation, and finally (4) summative comparative evaluation. In this article we first give some motivation and background for our methodology, then we describe each technique in some detail. We applied these techniques to a real-world battlefield visualization VE, as explained. Finally, we evaluate why this approach provides a cost-effective strategy for assessing and iteratively improving user interaction in VEs.

## Motivation

The user interaction components of VE applications are often poorly designed and rarely evaluated with users. The vast majority of VE research and design effort has gone into the development of visual quality and rendering efficiency. As a result, many visually compelling VEs are difficult to use and thus unproductive. While these VEs might make good entertainment applications, their usability problems prevent them from being useful for efficiently solving real-world problems.

Usability engineering[1] and user-centered design and evaluation[2] are newly emerging facets of VE development. VE designers and developers are becoming aware of traditional human-computer interface (HCI) usability efforts and beginning to apply and expand upon those methods for VEs. A few efforts have been reported to date; however, user-centered design and usability evaluation in VEs as a practice still lags. We have reached the point in VE development when we should shift from largely open-ended explorations of new technologies to more scientific studies of the benefits and impact of VEs on their users.

## The two development domains

Two distinct domains make up interactive system development—behavioral and constructional. The behavioral domain represents the view of the user and the user interaction with the application, while the constructional domain represents the view of the software developer and the overall system. The user interaction component is developed in the behavioral domain—the look and feel and behavior as a user interacts with an application. User interaction components include all icons, text, graphics, audio, video, and devices through which a user communicates with an interactive system, as well as locomotion, layout, content, and so on. The software component is developed in the constructional domain, including code for both the user interface and the rest of the application.

Roles that support each of these domains require different training, skills, and attitudes. While these roles are relatively well defined and the people holding them well trained for software development in the constructional domain—mainly for software and systems engineers—they're much less well defined and have far fewer well-trained practitioners for user interaction development in the behavioral domain. This holds especially true for usability engineering of VEs—very few experts exist in user interaction design and evaluation of VEs.

Thus, interaction designers and evaluators do their

0272-1716/99/$10.00 © 1999 IEEE

# Report Documentation Page

*Form Approved*
*OMB No. 0704-0188*

| | | |
|---|---|---|
| 1. REPORT DATE<br>**DEC 1999** | 2. REPORT TYPE | 3. DATES COVERED<br>**00-00-1999 to 00-00-1999** |

4. TITLE AND SUBTITLE
**User-Centered Design and Evaluation of Virtual Environments**

5a. CONTRACT NUMBER

5b. GRANT NUMBER

5c. PROGRAM ELEMENT NUMBER

6. AUTHOR(S)

5d. PROJECT NUMBER

5e. TASK NUMBER

5f. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
**Naval Research Laboratory,4555 Overlook Ave.**
**SW,Washington,DC,20375**

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)

10. SPONSOR/MONITOR'S ACRONYM(S)

11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT
**Approved for public release; distribution unlimited**

13. SUPPLEMENTARY NOTES

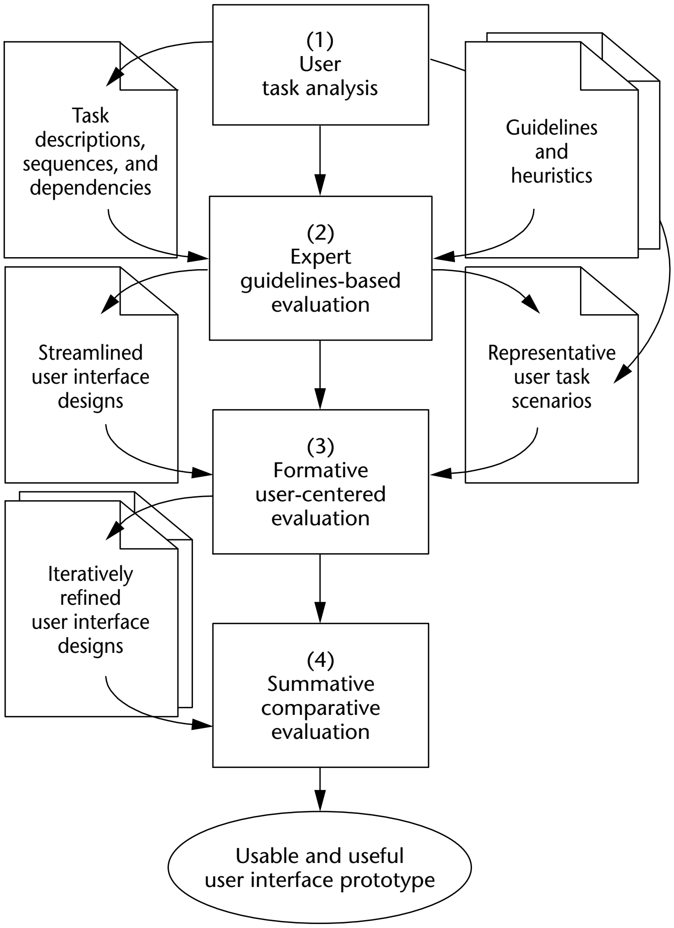
14. ABSTRACT

15. SUBJECT TERMS

| 16. SECURITY CLASSIFICATION OF: | | | 17. LIMITATION OF ABSTRACT | 18. NUMBER OF PAGES | 19a. NAME OF RESPONSIBLE PERSON |
|---|---|---|---|---|---|
| a. REPORT<br>**unclassified** | b. ABSTRACT<br>**unclassified** | c. THIS PAGE<br>**unclassified** | **Same as Report (SAR)** | **9** | |

work in the behavioral domain, while software and systems engineers and related roles do their work in the constructional domain. Well-known techniques from software engineering suit developing and evaluating the user interface software component. This kind of software evaluation can have many objectives, such as determining fidelity of a design to its implementation, reliability, reusability, and so on. Usability, however, is not one of these objectives, and usability engineering employs a very different set of methods. It isn't the user interface software component that's engineered for usability, but rather the user interaction component (which happens to be instantiated in software).

Cooperation between usability engineers and software engineers is essential for VEs to mature toward a truly user-centric work and entertainment experience. Thus, producing any interactive system, including a VE, requires both the behavioral and the constructional domains. Nonetheless, the domain that ensures usability, and in which usability engineering is applied, is the behavioral domain.



**1** **Methodology for the user-centered design and evaluation of VE user interaction.**

## Our methodology

VE researchers interested in applying proven usability design and evaluation methods discover few documented, well-tested methods for VE usability engineering. They often consider employing existing GUI-based evaluation and design methods, but limitations and incompatibilities between GUIs and VEs may render these methods inapplicable at best. Methods for usability engineering of VEs need to consider a broad variety of issues not addressed in current methods for evaluating usability of GUIs.

For example, how does an evaluator collect verbal protocol and interact with a user immersed in a virtual world that frequently generates its own sound and possibly even uses voice input to control the system? How can evaluators observe both users and visual scenes in a Cave Automated Virtual Environment (CAVE) without altering the users' sense of presence or situational awareness? How can we study, for example, the best way to represent a virtual person in a meeting (someone physically located elsewhere) to others in that meeting? How do preconceived notions and expectations of VE interfaces manifest themselves in subjective data, and how can we account for this manifestation?

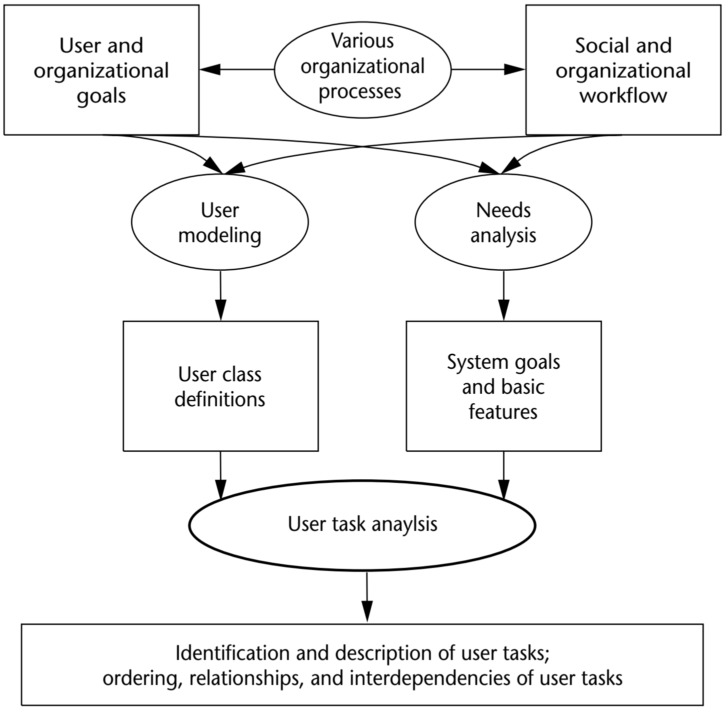
Other issues include, for example, how limits on observable data imposed by special VE equipment could impact usability engineering methods, by not allowing an evaluator to see a user's facial expression. Or a user's ability to move around, especially in a 3D VE, may make it more difficult for an evaluator to follow the user's actions to determine if that user is performing a specific task correctly.

To support rich and dynamic user-centered design and evaluation of VEs, we must forge new usability engineering methods that merge well-established techniques for evaluation and design of human activity with new, innovative methods capable of analyzing emerging VE-based interaction components. We've found a successful, cost-effective progression of methods for VE usability engineering that lets researchers not only improve VE usability, but address some of the pragmatic usability engineering questions presented above.

Our methodology, illustrated in Figure 1, is based on sequentially performing

1. user task analysis,
2. expert guidelines-based evaluation,
3. formative user-centered evaluation, and
4. summative comparative evaluations.

We describe each of these tasks in more detail below. While similar methodologies have been applied to traditional (GUI-based) computer systems, this particular methodology is novel because we specifically designed it for and applied it to VEs, and it leverages a set of heuristic guidelines specifically designed for VEs.


User and organizational goals
Various organizational processes
Social and organizational workflow
User modeling
Needs analysis
User class definitions
System goals and basic features
User task anaylsis
Identification and description of user tasks; ordering, relationships, and interdependencies of user tasks

2 A user task analysis identifies and describes user tasks as well as their ordering, relationships, and interdependencies.

## User task analysis

A user task analysis[3,4] is the process of identifying a complete description of tasks, subtasks, and methods required to use a system, as well as other resources necessary for user(s) and the system to cooperatively perform tasks. It follows a formal methodology, described in detail elsewhere.[3,4] As depicted in Figure 2, a user task analysis represents insights gained through an understanding of user, organizational, and social workflow; needs analysis; and user modeling. A user task analysis generates critical information used throughout all stages of the application development life cycle (and subsequently, all stages of the usability design and evaluation life cycle). A major result is a top-down decomposition of detailed user task descriptions for use by designers and evaluators. Equally revealing results include an understanding of required task sequences as well as sequence semantics. Thus, the results include not only the identification and description of tasks, but also information about the ordering, relationships, and interdependencies among user tasks.

Unfortunately, this critical step of user interaction development is often overlooked or poorly done. Without a clear understanding of user task requirements, both evaluators and developers must "best guess" or interpret desired functionality, which inevitably leads to poor user interaction design. Indeed, user interaction developers as well as user interface software developers claim that poor, incomplete, or missing user task analysis is one of the most common causes of poor user interaction design.

## Expert guidelines-based evaluation

Expert guidelines-based evaluation (heuristic evaluation or usability inspection) aims to identify potential usability problems by comparing a user interaction design—either existing or evolving—to established usability design guidelines. In this analytical evaluation, an expert in user interaction design assesses a particular interface prototype by determining what usability design guidelines it violates and supports. Then, based on these findings, especially the violations, the expert makes recommendations to improve the design. In the case of VEs, this proves particularly challenging because so few guidelines exist specific to VE user interaction.

Typically more than one person performs guidelines-based evaluations, since it's unlikely that any one person could identify all if not most of an interaction design's usability problems. Nielsen[5] recommends three to five evaluators for a GUI heuristic evaluation, since fewer evaluators generally cannot identify enough problems to warrant the expense, while more evaluators produce diminishing results at higher costs. It's not clear whether this recommendation is cost effective for VEs, since more complex VE interaction designs may require more evaluators than do GUIs.

Each evaluator first inspects the design independently of other evaluators' findings. Results are then combined, documented, and assessed as evaluators communicate and analyze both common and conflicting usability findings. Further, Nielsen[5] suggests a two-pass approach. During the first pass, evaluators gain an understanding of the general flow of interaction. During the second pass, evaluators identify specific interaction components and conflicts as they relate to both task flow and the larger-scoped interaction paradigm. This method is best applied early in the development cycle so that design issues can be addressed as part of the iterative design and development process.

Expert guidelines-based evaluations rely on established usability guidelines to determine whether a user interaction design supports intuitive user task performance.[5,6] While these heuristics are considered the de facto standard for GUIs, we have found them too general, ambiguous, and high level for effective and practical heuristic evaluation of VEs.

Recently, we produced a set of usability design guidelines specifically for VEs, contained within a framework of usability characteristics.[7] This framework document (available on the Web at http://www.vpst.org/jgabbard/ve/framework/) provides a reasonable starting point for heuristic evaluation of VEs. The complete document contains several associated usability resources, including specific usability guidelines, detailed context-driven discussion of the numerous guidelines, and citations of additional references.

The framework organizes VE user interaction design guidelines and the related context-driven discussion into four major areas:

1. users and user tasks,
2. input mechanisms,
3. virtual models, and
4. presentation mechanisms.

The framework categorizes 195 guidelines covering many aspects of VEs that affect usability, including locomotion, object selection and manipulation, user goals, fidelity of imagery, input device modes and usage, interaction metaphors, and more.

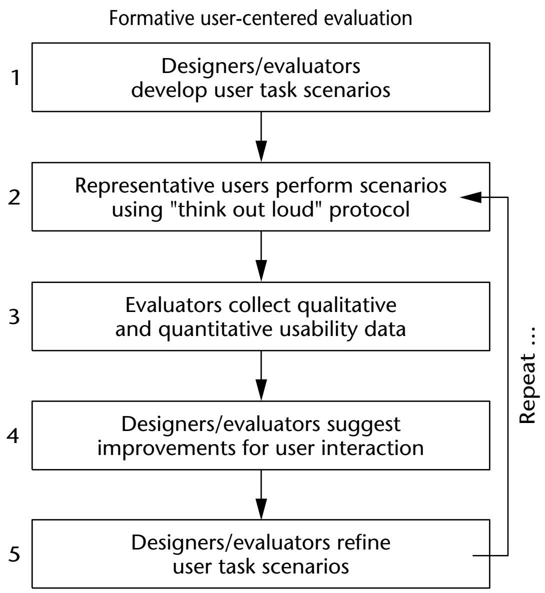
The guidelines presented within the framework document suit performing guidelines-based evaluation of VE user interfaces and interaction, since they provide broad coverage of VE interaction and interfaces yet are specific enough for practical application. For example, with respect to navigation within VEs, one guideline reads "provide information so that users can always answer the questions: Where am I now? What is my current attitude and orientation? Where do I want to go? How do I travel there?" Another guideline addresses methods to aid in usable object selection techniques, stating "use transparency to avoid occlusion during selection."

### Formative user-centered evaluation

Formative user-centered evaluation[3] is an empirical, observational evaluation method that ensures usability of interactive systems by including users early and continually throughout user interaction development. The method relies heavily on usage context (for example, user task, user motivation, and so on) as well as a solid understanding of human-computer interaction (and in the case of VEs, human-VE interaction). Therefore, a usability specialist generally proctors formative user-centered evaluations. Formative evaluation aims to iteratively and quantifiably assess and improve a user interaction design.

Figure 3 shows the steps of a typical formative evaluation cycle. The cycle begins with development of user task scenarios, which are specifically designed to exploit and explore all identified task, information, and work flows. Note that user task scenarios derive from results of the user task analysis. Moreover, these scenarios should provide adequate coverage of tasks as well as accurate sequencing of tasks identified during the user task analysis. Representative users perform these tasks as evaluators collect data. These data are then analyzed to identify user interaction components or features that both support and detract from user task performance. These observations are in turn used to suggest user interaction design changes as well as formative evaluation scenario and observation (re)design.

Note that in the formative evaluation process both qualitative and quantitative data are collected from representative users during their performance of task scenarios. Developers often have the false impression that usability evaluation has no "real" process and no "real" data. To the contrary, experienced usability evaluators collect large volumes of both qualitative data and quantitative data. Qualitative data are typically in the form of *critical incidents*,[3,8] which occur while a user performs task scenarios. A critical incident is an event that has a significant effect, either positive or negative, on user task performance or user satisfaction with the interface. Events that affect user performance or satisfaction therefore have an impact on usability. Typically, a critical incident is a problem encountered by a user (such as an error, being unable to complete a task scenario, or user confusion) that noticeably affects task flow or task performance. Quantitative data are generally related, for example, to how long it takes and the number of errors committed while a user performs task scenarios. These data are then compared to appropriate baseline metrics. Quantitative data generally indicate that a problem has occurred; qualitative data indicate where (and sometimes why) it occurred.

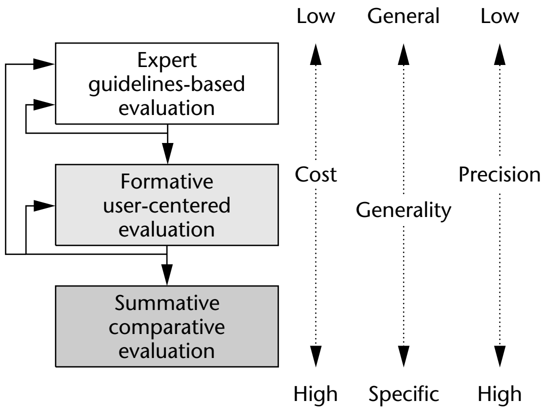

Formative user-centered evaluation
1
Designers/evaluators develop user task scenarios
2
Representative users perform scenarios using "think out loud" protocol
3
Evaluators collect qualitative and quantitative usability data
4
Designers/evaluators suggest improvements for user interaction
5
Designers/evaluators refine user task scenarios
Repeat ...

**3 Formative user-centered evaluation cycle.**

### Summative comparative evaluation

In contrast to formative user-centered evaluation, summative comparative evaluation[3] is an empirical assessment of an interaction design in comparison with other maturing interaction designs for performing the same user tasks. Summative evaluation is typically performed with some more-or-less final versions of interaction designs, and it yields primarily quantitative results. The purpose of summative comparative evaluation is to statistically compare user performance with different interaction designs, for example, to determine which one is better, where "better" is defined in advance.

When used to assess user interfaces, summative evaluation can be thought of as experimental evaluation with users comparing two or more configurations of user interface components, interaction paradigms, interaction devices, and so forth. Comparing devices and interaction techniques employs a consistent set of user task scenarios (developed during formative evaluation and refined for summative evaluation) resulting in primarily quantitative data results that compare (on a task by task basis) the designs' ability to support user task performance.


Expert guidelines-based evaluation
Formative user-centered evaluation
Summative comparative evaluation
Low
Cost
High
General
Generality
Specific
Low
Precision
High

**4 Additional properties of the expert guidelines-based, formative user-centered, and summative comparative evaluation methods.**

## An effective progression

Through our recent work, we found that the progression of methods we present suits cost-effective, efficient, design and evaluation of VEs particularly well.[9,10] Refer to Figure 1 throughout the following discussion.

A user task analysis provides the basis for design and evaluation in terms of what types of tasks and task sequences users will need to perform within a specific VE. This analysis generates (among other outputs) a list of detailed task descriptions, sequences, and relationships, user work, and information flow. It provides a basis for design and application of subsequent evaluation methods.

For example, the user task analysis may help eliminate or identify specific guidelines or sets of guidelines during expert guidelines-based evaluation. In a similar fashion, a user task analysis serves as both a basis for user evaluation scenario development as well as a checklist for evaluation coverage. That is, a well-developed task analysis provides evaluators with a complete list of end-use functionality detailing not only which tasks are to be performed but also likely task sequences and dependencies. Ordering and dependencies of user tasks is critical to powerful user evaluation scenario development. The closer the match between user task analysis and actual end user tasking, the better and more effective the final user interaction design.

An expert guidelines-based evaluation is the first assessment of an interaction design based on the user task analysis and application of guidelines for VE interaction design. This extremely useful evaluation removes many obvious usability problems from an interaction design. A VE interaction design expert will find both subtle and major usability problems through a guidelines-based evaluation. Once problems are identified, experts perform further assessment to understand how particular interaction components, devices, and so on affect user performance.

Results of expert guidelines-based evaluations are critical to effective formative and summative evaluations. For example, these results (coupled with results of user task analysis) serve as a basis for user scenario development. That is, if expert guidelines-based evaluation identifies a possible mismatch between implementation of a wireless 3D input device and manipulation of user viewpoint, then scenarios requiring users to manipulate the viewpoint should be included in formative evaluations.

Results of expert guidelines-based evaluations are also used to streamline subsequent evaluations. Further, critical usability problems identified during expert guidelines-based evaluation are corrected prior to performing formative evaluations, affording formative evaluations that don't waste time exposing those obvious usability problems addressed by the guidelines-based evaluation.

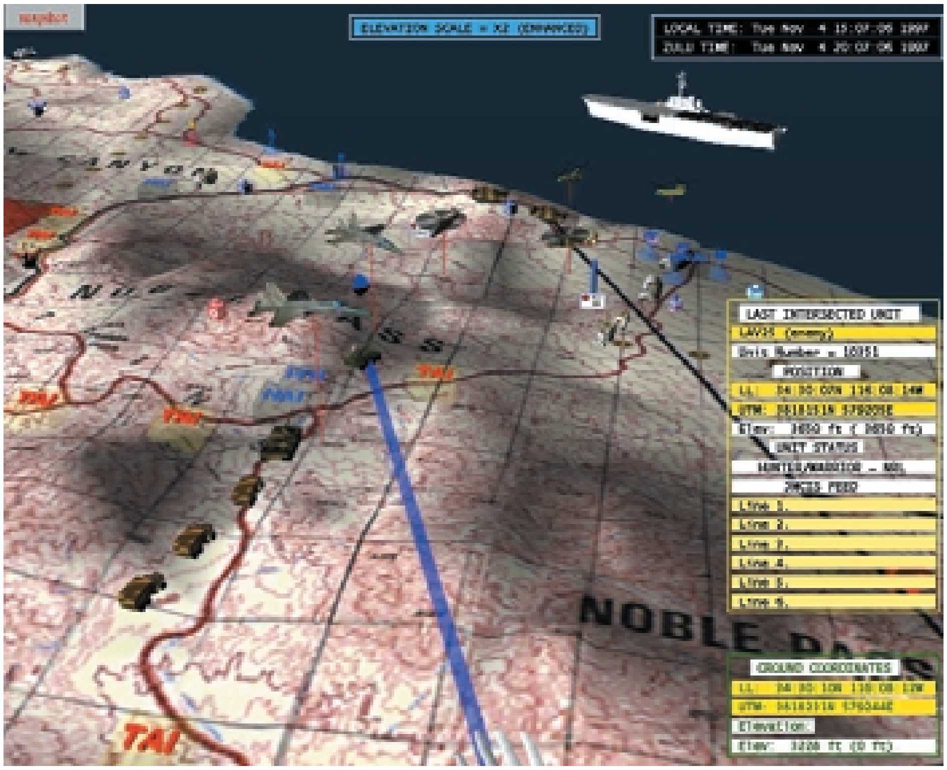
Because formative evaluation involves typical users, it most effectively uncovers issues (such as missing user tasks) that an expert performing a guidelines-based evaluation might be unaware of. A formative evaluation following a guidelines-based evaluation can focus not on major, obvious usability issues, but rather on those more subtle and more difficult to recognize issues. This becomes especially important because of the cost of VE development.

Coupling expert guidelines-based evaluations with formative user-centered evaluation helps successfully refine GUIs. Nielsen[5] recommends alternating expert guidelines-based evaluations and formative evaluation. The rationale is that no single method can reliably identify any and all usability problems. Indeed, guidelines-based evaluation and formative evaluation complement each other, often revealing usability problems that the other may have missed.[11]

Finally, a summative comparative evaluation following the preceding activities compares good apples to good oranges rather than comparing possibly rotten apples to good oranges. That is, summative studies comparing VEs whose interaction design has had little or no task analysis, guidelines-based evaluation, and/or formative evaluation may really be comparing one VE interaction design that is (for whatever reasons) inherently better—in terms of usability—to a different (and worse) VE interaction design. The first three methods produce a set of well-developed, iteratively refined, user interface designs. Subsequently, the designs compared in the summative study should be as usable, and comparably usable, as feasible. This means that any differences found in a summative comparison are much more likely the result of differences in the designs' basic nature rather than true differences in usability. Again, because of the cost of VE development, this confidence in results proves especially consequential.

The progression of methods is structured at a high level for application to any VE, regardless of the hardware, software, or interaction style used. Employing case-specific task analysis, guidelines, and user task scenarios facilitates broad applicability. As such, each specific method is flexible enough to support evaluation of any VE subsystem (visual, auditory, or haptic, for example) or combination thereof.

Figure 4 shows additional properties of the three types of evaluation. The solid arrows underscore the methods' application sequence. We recommend applying expert guideline-based evaluation first, perhaps iterating several times. The least expensive evaluation to perform and very general, it can cover large portions (if



ELEVATION SCALE = X2 (ENHANCED)

LOCAL TIME: Tue Nov 4 15:07:05 1997
ZULU TIME: Tue Nov 4 20:07:05 1997

LAST INTERSECTED UNIT
LAV25 (enemy)
Unit Number = 10051
POSITION
LL: 34 30 07N 116 08 14W
UTM: 36183131N 3780098E
Elev: 3050 ft ( 3050 ft)
UNIT STATUS
HUNTER/WARRIOR – NBL
JMCIS FEED
Line 1.
Line 2.
Line 3.
Line 4.
Line 5.
Line 6.

GROUND COORDINATES
LL: 34 30 10N 116 08 12W
UTM: 36183131N 3780544E
Elevation:
Elev: 3208 ft (0 ft)

**5** A screen shot of the Dragon battlefield visualization VE.

not all) of the user interface. However, expert guideline-based evaluation isn't very precise: it gives only general indications of what might be wrong and doesn't address how to fix usability problems.

We next apply formative usability evaluation, which is more expensive (it requires users and task scenarios) and less general (a smaller portion of the user interface can be covered per session). However, the results are more precise, often revealing where problems occur and suggesting ways to fix them. Typically iterated several times, formative usability evaluation may lead to additional expert guidelines-based evaluation of modified or missed portions of the user interface.

Finally, summative evaluations are very expensive (requiring many more subjects than formative usability evaluations) and also extremely specific—they can answer only very narrowly defined questions. However, summative evaluations answer these questions with a high degree of precision: it's the only type of evaluation that can statistically quantify how much better one design is than another.

## The Dragon battlefield visualization VE

Collaborating with researchers from Virginia Tech, personnel at the Naval Research Laboratory's Virtual Reality Lab developed a VE for battlefield visualization called Dragon (Figure 5).[12] We applied a slightly less refined version of our usability engineering methodology to the design and evaluation of Dragon's user interaction component. In this section we briefly describe Dragon and the application domain of battlefield visualization. In the next section we discuss how we applied the methodology to Dragon.

For decades, battlefield visualization has relied on paper maps of the battlespace placed under sheets of acetate. As intelligence reports arrive from the field, technicians use grease pencils to mark new information on the acetate. Commanders then draw on the acetate to plan and direct various battlefield situations. Thus, the map and acetate together present a visualization of the battlespace. Using maps and overlays can take several hours to print, distribute, and update. Historically (before high-quality paper maps), these same operations were performed on a sandtable (a box filled with sand shaped to replicate the battlespace terrain). Commanders moved small physical replicas of battlefield objects to direct battlefield situations. Currently, the fast-changing modern battlefield produces so much time-critical information that these cumbersome, time-consuming methods are inadequate for effectively visualizing the battlespace.

In Dragon, a Responsive Workbench provides a 3D display for observing and managing battlespace information shared among commanders and other battle planners. Visualized information includes a high-resolution terrain map; entities representing friendly, enemy, unknown, and neutral units; and symbology representing other features such as obstructions or key battle objectives. Dragon receives electronic intelligence feeds that provide constantly updated, displayable information about each entity's status, including position, speed, heading, damage condition, and so forth. Users can navigate to observe the map and entities from any angle and orientation, and can query and manipulate entities.

A user interacts with Dragon using a three-button game flightstick (removed from its base) fitted with a six-degrees-of-freedom position sensor. Dragon tracks the flightstick's position and orientation relative to an emitter located on the front center of the Workbench. A virtual laser pointer metaphor is used: a laser beam appears to come out of the flightstick, allowing interaction with the terrain or object that the beam intersects.

## Applying the methodology to Dragon

We used the basic Dragon VE application as an instrumentable testbed, modified as needed for our expert guidelines-based and formative user-centered evaluation purposes. We performed extensive evaluations over a nine-month period, using anywhere from one to three users for each cycle of evaluation, and using two to three evaluators per session. From a single evaluation session, we often uncovered design problems so serious that it was pointless to have different users attempt to perform the scenarios with the same design. So we would iterate the design, based on our observations, and begin a new cycle of evaluation. We went through four major cycles of iteration during our evaluation of Dragon,[9] each cycle using the progression of usability methods described previously.

### *User task analysis*

Early Dragon developers performed a user task analysis by interviewing several US Navy personnel who use the current system of battlespace visualization (acetate, paper maps, and grease pens). This included both commanders and lower-level technicians. Important Dragon-specific tasks identified included planning and shaping a battlefield, comprehending situational awareness in a changing battlespace, performing engagement and execution exercises, and carrying out "what if" (contingency planning) exercises. The user task analysis also examined how personnel perform their current battlefield visualization tasks. This task analysis took place before we joined the project. However, we revisited the task analysis several times during the course of our own early work and enhanced it with our own observations and interviews.

During our early work, we observed that locomotion—how users manipulate their viewpoint to move from place to place in a virtual world (in this case, the map for battlefield visualization)—profoundly affects all other user tasks. If a user cannot successfully locomote in a virtual world, then other user tasks (involving specific objects or groups of objects, for example) become impossible. A user cannot query an object if the user cannot navigate through the virtual world to get to that object. Locomotion is a generic (as opposed to Dragon-specific) task that users of almost any VE will have to perform. Thus, we chose locomotion as a major focus of our subsequent work with Dragon.

### *Expert guidelines-based evaluations*

During our expert guidelines-based evaluations, various user interaction design experts worked alone or collectively to assess the evolving user interaction design for Dragon. In our earliest heuristic evaluations, the experts didn't follow specific user task scenarios per se, but simply engaged with the user interface. All experts knew enough about the purpose of Dragon as a battlefield visualization VE to explore the kinds of tasks most important for users. During each heuristic evaluation session, one person typically "drove," holding the flightstick and generally deciding what and how to explore in the application. One and sometimes two other experts observed, commented, and collected data. Much discussion occurred during each session.

We were often, but not always, the experts assessing the current design. Our assessment and discussions were guided largely by our own knowledge of interaction design for VEs and, more formally, by the framework for usability characteristics[7] discussed above. This framework provided a more structured means of evaluation than merely wandering around in the application. It also provided guidance on how to make modifications to improve discovered design guideline violations.

Major design problems uncovered by the expert guidelines-based evaluation included poor mapping of locomotion tasks (pan, zoom, pitch, heading) to flightstick buttons, missing user tasks (exocentric rotate, terrain following), problems with damping of map movement in response to flightstick movement, and inadequate graphical and textual feedback to the user about the current locomotion task (pan, zoom, and so forth). We discuss these problems, and how we addressed them, elsewhere.[9] After our cycles of expert guidelines-based evaluation had revealed and remedied as many design flaws as possible, we moved on to formative evaluations.

### *Formative user-centered evaluations*

Based on our user task analysis and early expert guidelines-based evaluations, we created a set of user task scenarios consisting of benchmark user tasks, carefully considered for coverage of specific issues related to locomotion. For example, some of the tasks exploited an egocentric (users move themselves) locomotion metaphor, while others exploited an exocentric (users move the world) locomotion metaphor. Some scenarios exercised various locomotion tasks (degrees of freedom: pan, zoom, rotate, heading, pitch, roll) throughout the virtual map world. Other scenarios served as primed exploration or nonprimed searches, while still others were designed to evaluate rate control versus position control in the virtual world. We thoroughly pretested and debugged all scenarios before presenting them to users during an evaluation session.

During each of six formative evaluation sessions, we followed a formal protocol of welcoming the user, giving an overview of the evaluation about to be performed, and then explaining the Responsive Workbench and the Dragon application. We carefully avoided explaining details of the Dragon interaction design, since that was what we were evaluating. Then we asked the user to play with the flightstick to figure out which button activated which locomotion task (pan, zoom, and so on). We timed each user as they attempted to determine this and took notes on comments they made and any critical incidents that occurred. Once a user had successfully figured out how to use the flightstick, we began having them perform the scenarios. If about 15 minutes passed without a user figuring out the flightstick and its buttons (this happened in only one case), we filled in details that they had not yet determined and moved on to scenarios.

Time to perform the set of scenarios ranged from about 20 minutes to more than an hour. We timed user performance of individual tasks and scenarios, and counted errors they made during task performance (quantitative data). A typical error was moving the flightstick in the wrong direction for the particular locomotion metaphor (exocentric or egocentric) currently in use. Other errors involved simply not being able to maneuver the map (to rotate it, for example) and persistent problems with mapping locomotion tasks to flightstick buttons. (Again, we discuss these further elsewhere.[9]) We also carefully noted critical incidents, especially related to errors, and constructive comments users made about the design (qualitative data).

During each session, we had at least two and often three evaluators present. The leader ran the session and interacted with the user; the other one or two evaluators recorded timings, counted errors, and collected qualitative data. While both the expert guidelines-based evaluation sessions and the formative evaluation sessions were personnel-intensive (with two or three evaluators involved), we found that the quality and amount of data collected by multiple evaluators greatly outweighed the cost of those evaluators. After each session, we analyzed both the quantitative and qualitative data, and based the next design iteration on our results.

### *Summative comparative evaluations*

Our current work aims to summatively evaluate the mature locomotion design. During our expert guidelines-based and formative evaluations, we discovered many different variables affecting locomotion usability in VEs. We narrowed this (initially large) list to five variables, based on the framework of usability characteristics,[7] our observations during heuristic and formative evaluations, and our expertise in VE interaction design. We feel these five variables have the greatest effect on locomotion and are therefore the most important candidates for summative evaluations:

1. locomotion metaphor (ego- versus exocentric),
2. gesture control (controls rate versus controls position),
3. visual presentation device (workbench, desktop, CAVE),
4. head tracking (present versus not present), and
5. stereopsis (present versus not present).

## Lessons learned

As explained, we found that our usability engineering methodology had a major impact: Results from formative usability evaluations inform the design of summative studies by helping determine appropriate usability characteristics to evaluate and compare in summative studies. Invariably, numerous alternatives can be considered as factors in a summative evaluation. Formative evaluations typically point out the most important usability characteristics and issues (such as those that recur most frequently, those that have the largest impact on user performance and satisfaction, and so on). These issues then become strong candidates for inclusion in a summative evaluation.

For example, if formative evaluation shows that users have a problem with format or placement of textual information in a heavily graphical display, a summative evaluation could explore alternative ways of presenting such textual information. Further, if users want different display modes (for example, stereoscopic and monoscopic, head-tracked and static, landscape view and overhead view of a map), these various configurations can also be the basis of rich comparative studies related to usability. As yet another example of a potential usability problem, users might have difficulty moving around in an immersive 3D version of a VE, but not in a 2D, nonimmersive version. A summative study could investigate what parameter(s) of the 3D version causing the problem don't appear in the 2D version.

An important advantage of applying the complete progression of methods is the timeliness of assessment efforts, aligning each component's strengths (such as level of detail or breadth of focus) with concurrent efforts in the software development process. For example, a user task analysis typically is performed at the onset of interaction design, prior to any prototype development. As prototype designs (paper and pencil prototypes, for example) start to emerge, expert guidelines-based evaluation can begin. As computer-based prototypes are developed, they take on a richer set of functionality, perfect for iterative formative user-centered evaluation. Finally, one or more candidate designs are available for summative comparative evaluation. Once complete, results and documentation from evaluation efforts provide an effective means of persistent design rationale. In complex development environments, tracking—often months after the fact—why particular interaction design changes were made can be very difficult if not impossible.

To ensure accuracy and aid effectiveness, the design and development team should include one or more domain experts. These experts provide specific context-related information to help usability experts understand cognitive task and information flow requirements. Domain experts also help direct and rank analysis foci so that evaluation resources are allocated to the most important usage issues. Moreover, having a domain expert on-board early in the design, evaluation, and development cycles helps that expert understand the domain of usability evaluation. This enables the domain expert to become a much more effective resource during subsequent evaluation phases.

This concludes our presentation of a methodology for usability engineering of virtual environments. We hope this work provides a starting point for techniques that let practitioners engineer VE interaction that is both useful and usable. ■

## Acknowledgments

Many people contributed to Dragon's development and therefore to this reported work. NRL personnel involved with major efforts include Jim Durbin, Tony

King, Eddy Kuo, Brad Colbert, and Chris Scannell. Other developers included John Crowe, Josh Davies, Bob Doyle, Rob King, Greg Newton, and Josh Summers. At NRL, Larry Rosenblum and Dave Tate gave leadership, inspiration, and guidance to this project. Jim Templeman and Linda Sibert of NRL and Bob Williges of Virginia Tech provided valuable suggestions. This research was funded by the Office of Naval Research under program managers Helen Gigley and Paul Quinn. We would like to thank Helen Gigley for her continued support of an ongoing synergistic collaboration in human-computer interaction research between Virginia Tech and NRL over the past several years.

## References

1. J. Nielsen, *Usability Engineering*, Academic Press, San Diego, Calif., 1993
2. D.A. Norman, and S.W. Draper, eds., *User Centered System Design*, Lawrence Erlbaum Associates, Hillsdale, N.J., 1986.
3. D. Hix and H.R. Hartson, *Developing User Interfaces: Ensuring Usability through Product and Process*, John Wiley and Sons, New York, 1993.
4. J.T. Hackos and J.C. Redish, *User and Task Analysis for Interface Design*, John Wiley and Sons, New York, 1998.
5. J. Nielsen, "Heuristic Evaluation," in *Usability Inspection Methods*, John Wiley and Sons, New York, 1994, pp. 25-62.
6. J. Nielsen and R. Molich, "Heuristic Evaluation of User Interfaces," *Proc. ACM CHI 90 Conf.,* ACM Press, New York, April 1990, pp. 249-256.
7. J.L. Gabbard, *A Taxonomy of Usability Characteristics in Virtual Environments*, master's thesis, Dept. of Computer Science and Applications, Virginia Polytechnic Institute and State University, 1998, http://www.theses.org/vt.htm.
8. E.M. del Galdo et al., "An Evaluation of Critical Incidents for Software Documentation Design," *Proc. 30th Annual Human Factors and Ergonomics Society Conf.*, Human Factors and Ergonomics Society, Santa Monica, Calif., 1986, pp. 19-23.
9. D. Hix et al., "User-Centered Design and Evaluation of a Real-Time Battlefield Visualization Virtual Environment," *Proc. IEEE Virtual Reality 99 Conf.*, IEEE CS Press, Los Alamitos, Calif., 1999, pp. 96-103.
10. J.L. Gabbard et al., "Usability Evaluation Techniques: A Novel Method for Assessing the Usability of an Immersive Medical VE," *Proc. Virtual Worlds and Simulation Conf.* (VWSIM 99), Society for Computer Simulation Int'l, San Diego, Calif., 1999, pp. 165-170.
11. H.W. Desurvire, J.M. Kondziela, and M.E. Atwood, "What Is Gained and Lost when Using Evaluation Methods other than Empirical Testing," *People and Computers VII*, Cambridge University Press, Cambridge, UK, 1992, pp. 89-102.
12. J. Durbin et al., "Battlefield Visualization on the Responsive Workbench," *Proc. IEEE Visualization 98*, IEEE CS Press, Los Alamitos, Calif., 1998, pp. 463-466.

***Joseph L. Gabbard*** *is the lead scientist at VPST, where he performs VE-based human-computer interaction research. He is currently interested in researching and developing usability engineering methods specifically for VEs. Other interests include developing innovative and intuitive interaction techniques employing ubiquitous input technology. He is currently pursuing his PhD in computer science at Virginia Polytechnic Institute in Blacksburg, Virginia. He received his MS in computer science, BS in computer science, and BA in sociology from Virginia Tech. He is a member of the IEEE, IEEE Computer Society, and Society for Computer Simulation International.*

***Deborah Hix*** *is a Computer Science faculty member at Virginia Polytechnic Institute and State University in Blacksburg, Virginia, and a founder and principal investigator of the Virginia Tech Human-Computer Interaction (HCI) Project. Most recently, Hix has extended her HCI work into VE usability. She has done extensive consulting and training in the area of user interface development for nearly 20 years. She is co-author of Developing User Interfaces: Ensuring Usability through Product and Process (John Wiley and Sons, New York, 1993).*

***J. Edward Swan II*** *is a scientist with the Virtual Reality Laboratory at the Naval Research Laboratory, where he conducts research in computer graphics and human-computer interaction. At the Naval Research Laboratory he is primarily motivated by the problem domain of battlefield visualization. Currently he is studying effective VE locomotion techniques for battlefield visualization, as well as new techniques in terrain rendering. He received his BS from Auburn University, and his MS and PhD from Ohio State University in 1997. He is a member of ACM, Siggraph, Sigchi, IEEE, and the IEEE Computer Society.*

*Readers may contact Gabbard at VPST, 2000 Kraft Dr., Suite 2600, Blacksburg, VA 24060-6354, e-mail jgabbard@vpst.org, http://www.vpst.org.*